

2

Branching in order to iterate

Creating options to consider for better outcomes

Why branching?

Agile culture is about teams being able to work loosely coupled, allowing individual team members to iterate on parts of key tasks to individually experiment, then showing these experiments to other team members for comment and then identifying and integrating the best strains of individual iteration to improve the overall shared work of a team.

GitHub is a purpose-built platform to break down work into issues and then socially collaborate on each issue to improve the units of work and sustain or improve team velocity in each sprint.

Since most of the work in this course is being done in GitHub through open issues, the main repository (default, or master branch) has the backlog of work (stories). Branching allows a parallel branch to be used to try out new ideas with the intent at some point to submit them to be considered to be merged back into the master branch. There can be many parallel branches in progress (your branch, and other branches) and all who have access to a repository can see all branches.

How branching works

1.

Creating a Branch

When you create a branch in your project, you're creating an environment where you can try out new ideas with the intent to merge the branch with the master branch.

Changes you make on a branch don't affect the master branch, so you're free to experiment and commit changes, safe in the knowledge that your branch won't be merged until it's ready to be reviewed by someone you're collaborating with.

2.

Adding Commits

Whenever you add, edit, or delete a file, you're making a commit, and adding them to your branch. This process of adding commits keeps track of your progress as you work on a feature branch.

Commits also create a transparent history of your work that others can follow to understand what you've done and why - otherwise when you create a pull request, the person who reviews your work will not know why.

3.

Opening & Submitting a Pull Request

Submit a pull request to the project owner about the changes you'd like them to consider and discuss about your commits. Pull Requests are designed to encourage and capture this type of conversation.

By incorporating certain keywords into the text of your Pull Request, you can associate issues with code.

A pull request can be accepted to be deployed into the main branch or closed (not accepted).

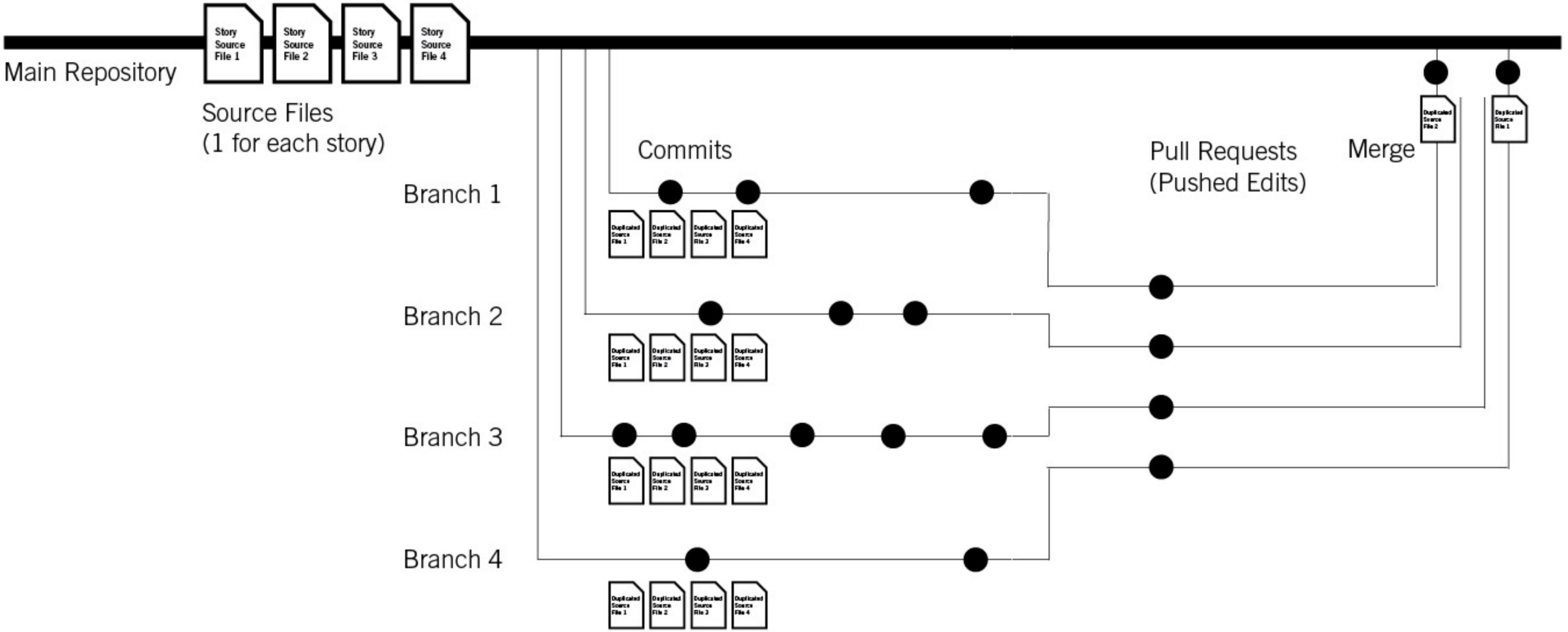
4.

Merging & Deploying

Once your pull request has been reviewed and the branch passes your tests, you can deploy your changes to verify them in production.

If your branch causes issues, you can roll it back by deploying the existing master into production. If a pull request is accepted, it is merged into the master branch.

How branching works



What is in a branch?

When a branch is created, in the GitHub code tab, two files are automatically created : a general license file and a readme file.

A branch owner can create a new file, name it, and write content within it. Every action that is done within a file or any new files that are saved is called a “commit.” An owner can also upload additional files (PDF, JPEG, etc.) in the branch to support what is in a new file. All of these files are “work” to iterate on a specific issue.

|